International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 1, Jan-Feb 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Optimize Resource Utilization using Enhanced Job Scheduling Algorithm in Cloud Computing Environments

Vipin Mishra[1], Dr. Mohit Singh Tomar[2], Dr. Ritu Shrivastava[3]

[1]Research Scholar, [2]Associate Professor, [3]Head and Professor,
[1,2,3]Department of CSE, SIRT, Bhopal, Madhya Pradesh, India

ABSTRACT

In a distributed framework, conveyed figuring represents an innovative and cost-effective domain. It facilitates a substantial shift in the environment that enhances compensation for representation while considering customer needs. The cloud represents a collaborative assembly of innovative techniques that merge to create a unified computing system while limiting its scope. Facilitating consent to widely spread and remote resources is the core objective of distributed care. The cloud is encountering a significant range of challenges, particularly in organization, and is taking incremental steps towards transformation. Booking scheduling encompasses a set of techniques aimed at optimizing the management of tasks that a computer system must execute. As circumstances evolve and such initiatives progress, the scheduler modifies the approach to how tasks are organized. The proposed approach enhances the estimation of Assignment Movement Combination Planning for expert task execution and evaluation through the use of FCFS and Least Fulfilment Time Booking. This process eliminates the from-line list and reverts to the foundational machine through management after solidifying with the garnish off, which requires a significant amount of time. The procedure operates within the tool repository for CloudSim 3.0.1, structured in NetBeans 8.1. The result shows that it offers a more innovative approach that stands apart from straightforward, clear-cut assessment of reservations. When examining FCFS and MCT on their own, the asset utilization rates are recorded at 0.52% and 11.45%, respectively.

KEYWORDS: *Distributed Framework, Innovative Approach, CloudSim 3.0.1, NetBeans 8.1, MCT, FCFS*

How to cite this paper*:* Vipin Mishra | Dr. Mohit Singh Tomar | Dr. Ritu Shrivastava "Optimize Resource Utilization using Enhanced Job Scheduling Algorithm in Cloud Computing Environments" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-1, February 2025, pp.969-975, URL: www.ijtsrd.com/papers/ijtsrd75145.pdf

IJTSRD75145

1. INTRODUCTION

In cloud computation environments, the demand for resources is constantly increasing, and efficiently managing these resources is crucial for optimizing performance and reducing operational costs. Job scheduling plays a vital role in achieving this by allocating cloud resources to tasks in an effective manner. As cloud systems scale, traditional scheduling algorithms often struggle to meet the performance requirements, leading to inefficient resource utilization, increased processing time, and higher energy consumption.

The paper "Optimize Resource Utilization using Enhanced Job Scheduling Algorithm in Cloud Computing Environments" presents a novel approach to address these challenges. It proposes an enhanced job scheduling algorithm that improves the allocation of resources across different jobs, aiming to optimize system performance and minimize inefficiencies. The enhanced algorithm focuses on dynamic load balancing, task prioritization, and efficient resource management. By considering factors such as resource availability, task performance time, and the energy feasting of various cloud resources, the proposed algorithm ensures that resources are utilized more effectively, resulting in reduced processing time and cost savings.

Furthermore, the proposed solution incorporates intelligent decision-making mechanisms that allow the system to adapt to changing workloads and dynamically adjust scheduling strategies. This

flexibility ensures that cloud resources are utilized to their full potential, even during periods of peak demand. By reducing task delays, minimizing idle times, and ensuring better allocation of computing, storage, and networking resources, the enhanced job scheduling algorithm can significantly improve the overall efficiency of cloud computing environments.

In summary, this study highlights the standing of job scheduling in cloud systems and introduces a more effective algorithm that addresses the limitations of existing methods. By optimizing resource utilization, the proposed solution not only enhances cloud service performance but also contributes to a more sustainable and cost-effective cloud infrastructure.

2. Related Work

In recent years, cloud computing has evolved to become a critical enabler of scalable, flexible, and cost-effective computing resources. The rapid expansion of cloud services has necessitated efficient resource management to maximize the performance and cost-effectiveness of these services. A significant area of research in cloud computing is the optimization of resource utilization, with job scheduling algorithms being a central focus. The goal is to efficiently allocate cloud resources (e.g., compute power, memory, and storage) to jobs or tasks while minimizing costs and improving Quality of Service (QoS). Enhanced job scheduling algorithms are emerging as a promising solution to these challenges, with various approaches focused on improving resource utilization, reducing job completion time, and ensuring load balancing across the system. The review of recent literature highlights several innovative strategies aimed at addressing the complexities inherent in scheduling tasks within the dynamic, distributed, and virtualized environment of cloud computing.

A major challenge in cloud computing is the efficient allocation of resources under varying workloads, which require job scheduling algorithms that can respond to dynamically changing system conditions. Several studies focus on developing enhanced job scheduling algorithms that take into account factors such as task dependencies, resource availability, and task priority. One notable approach is based on metaheuristic optimization methods, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), which have been widely applied to optimize job scheduling by balancing workload distribution and minimizing resource contention (Zhou et al., 2022). These algorithms attempt to find near-optimal solutions by exploring the solution space through iterations and local search processes. Metaheuristics have shown to effectively address the complexities of large-scale cloud environments by offering flexibility in finding solutions that traditional approaches may not uncover.

Another promising method is the use of machine education techniques to improve the accurateness and efficiency of job scheduling. Several studies have proposed intelligent job schedulers that leverage machine learning models to predict resource demand, job completion time, and optimal scheduling order. For example, deep reinforcement learning (DRL) has been applied to dynamically allocate resources in cloud environments (Liu et al., 2021). DRL models continuously learn and adjust the scheduling policy based on the system’s feedback, enabling them to optimize resource allocation based on the observed performance. This self-learning capability allows for better adaptation to unpredictable workloads, improving resource utilization over time. In a similar vein, other researchers have explored hybrid models that combine machine learning with heuristic methods to enhance scheduling decisions (Hussain et al., 2021). These models combine the strengths of both approaches, achieving improved performance compared to traditional job schedulers.

Task offloading is another area where resource utilization can be optimized through enhanced job scheduling algorithms. Offloading tasks to specialized resources or edge computing platforms has gained attention as a means to reduce latency and increase throughput, especially in resource-constrained environments (Singh et al., 2022). Research in this domain has explored hybrid job scheduling algorithms that not only optimize local cloud resource allocation but also intelligently distribute tasks across different layers of the computing architecture, including edge and fog computing platforms. These algorithms aim to reduce communication overheads and mitigate the impact of network delays, contributing to more efficient resource usage in multi-tier cloud systems. In particular, the integration of fog and edge computing into cloud environments offers a decentralized scheduling model, allowing for more localized task execution, thereby improving resource utilization and minimizing energy consumption.

Additionally, load complementary techniques play a critical role in optimizing supply utilization in cloud environments. Unbalanced workloads can result in underutilized or overburdened resources, leading to inefficiencies and degraded system performance. Recent literature has highlighted the importance of dynamic load balancing strategies that redistribute tasks based on real-time system conditions, such as resource availability, network congestion, and task complexity (Rao et al., 2023). Some researchers have

proposed using game theory to develop load balancing algorithms that encourage fair resource distribution among multiple cloud nodes, thereby optimizing overall resource utilization. These game-theoretic approaches allow for self-organizing systems where participants (i.e., cloud nodes) adjust their behavior to achieve optimal resource allocation without the need for centralized control (Zhang et al., 2023).

The integration of cloud-based job scheduling with energy-efficient resource management has also received significant attention. Researchers have explored scheduling strategies that aim to minimize energy consumption while ensuring efficient resource allocation. One such approach is Dynamic Voltage and Frequency Scaling (DVFS), which regulates the power consumption of cloud servers based on assignment demands. In combination with job scheduling algorithms, DVFS helps balance performance and energy efficiency by reducing unnecessary power consumption during low-demand periods (Babar et al., 2022). Other energy-aware scheduling algorithms aim to minimize carbon footprints by considering environmental constraints and aligning resource utilization with green computing principles (Kumar & Soni, 2023).

A particularly promising trend in recent literature is the development of multi-objective optimization techniques for job development in cloud situations. These techniques optimize manifold conflicting purposes simultaneously, such as minimizing execution time, reducing energy consumption, and maximizing resource operation. Pareto-based methods, which provide a set of optimal solutions rather than a single solution, are frequently employed in this context. Researchers have demonstrated that multi-objective job scheduling algorithms can effectively balance trade-offs between conflicting objectives, leading to more comprehensive solutions that are applicable to a wide range of cloud computing scenarios (Bose et al., 2021). These algorithms incorporate various performance metrics into the optimization process, such as task completion time, resource utilization, and system reliability.

The ongoing evolution of cloud computing and the increasing demand for real-time processing and heterogeneous resources have led to a surge of innovative scheduling algorithms that aim to tackle the complex challenges of resource optimization. From metaheuristics and machine learning-based methods to energy-efficient strategies and multi-objective optimization, recent studies reveal a broad spectrum of solutions aimed at improving resource utilization. These developments underline the importance of continually advancing scheduling algorithms in order to meet the growing demands of cloud computing environments.

3. Problem Identification

The obvious problem with the current work is as follows:

- Low Asset usage: a significant portion of the certified energy usage is often attributed to the liveliness consumption of underutilized assets. For a successful climate in the cloud, asset utilization should be enhanced for areas of strength.
- High Makespan: The High QoS Essential work is a development to the low QoS center task. Since the objective of profession reservation is to boundary costs by diminishing makespan period, the client has sufficient cash for the work spaces, which are not definitively subject to use time.
- High Execution Rate: The typical execution period, which demonstrations the quantity of endeavors, should be feasible inside a detailed time period. A high implementation cost shows a disappointing getting sorted out technique.

4. Research Objectives

The objectives of this piece of art are as follows:

- Most notable usage of assets: Develop asset use to substantially reduce energy consumption.
- Reducing the immovable makespan and increasing the usage of virtual machines are two ways to reduce the makespan of occupation movements. The problem of errand booking is as fast and dirty as the problem of multi-target improvement.
- To reduce execution costs: Careful coordination of the booking work process executions is necessary to minimize the full-scale execution costs of the asset utilization.

5. Methodology

The following is the fundamental algorithm of the suggested technique.

First Step: Enter the list of tasks together with their size, execution duration, and resource allocations made using the suggested scheduling method. Examine the following list of jobs:

J1(1,4), J2(2,2), J3(2,4), J4(3,4), J5(1,10), J6(1,6), J7(2,4), J8(5,2), J9(4,2), J10(1,6).

Consider to resources R1 and R2 with their node size

Nodes

Resources	P1	P2	P3	P4	P5
R1					
R2					

Step 2: Time interval size is 2 Sec. At time T=0

	P1	P2	P3	P4	P5
R1	J1	J2	J2	J3	J3
R2	N	J4	J4	J4	J5

Queue: J6(1,6), J7(2,4), J8(5,2), J9(4,2), J10(1,6)

Step 3: At time T=2

(a)

	P1	P2	P3	P4	P5
R1	J1			J3	J3
R2	N	J4	J4	J4	J5

Queue : J6(1,6),J7(2,4), J8(5,2), J9(4,2), J10(1,6)

(b)

	P1	P2	P3	P4	P5
R1	J1	J5		J3	J3
R2	N	N			

Queue : J4(3,2), J6(1,6), J7(2,4), J8(5,2), J9(4,2), J10(1,6)

(c)

	P1	P2	P3	P4	P5
R1	J1	J5	J6	J3	J3
R2	N	N	N		

Queue: J4(3,2), J7(2,4), J8(5,2), J9(4,2), J10(1,6)

(d)

	P1	P2	P3	P4	P5
R1	J1	J5	J6	J3	J3
R2	N	N	N	J7	J7

Queue: J4(3,2), J8(5,2), J9(4,2), J10(1,6)

Step 4: At time T=4

(a)

	P1	P2	P3	P4	P5
R1		J5	J6		
R2		N	N	J7	J7

Queue: J4(3,2), J8(5,2), J9(4,2), J10(1,6)

(b)

	P1	P2	P3	P4	P5
R1	J4	J5	J6	J4	J4
R2		N	N	J7	J7

Queue: J8(5,2), J9(4,2), J10(1,6)

(c)

	P1	P2	P3	P4	P5
R1	J4	J5	J6	J4	J4
R2	J10	N	N	J7	J7

Queue: J8(5,2), J9(4,2)

Step 5: At time T=6

(a)

	P1	P2	P3	P4	P5
R1		J5	J6		
R2	J10	N	N		

Queue: J8(5,2), J9(4,2)

(b)

	P1	P2	P3	P4	P5
R1	J10	J5	J6		
R2	N	N	N		

Queue: J8(5,2), J9(4,2)

Step 6: At time T=8

	P1	P2	P3	P4	P5
R1	J10	J5			
R2	N				

Queue: J8(5,2), J9(4,2)

Step 7: At time T=10

(a)

R1					
R2					

Queue: J8(5,2), J9(4,2)

(b)

	P1	P2	P3	P4	P5
R1	J8	J8	J8	J8	J8
R2					

Queue: J9(4,2)

(c)

	P1	P2	P3	P4	P5
R1	J8	J8	J8	J8	J8
R2	J9	J9	J9	J9	

Step 8: Total makespan of given queue is 10

6. Results and Analysis

In order to set up the simulation in accordance with standard CloudSim 3.0.2, the Main period generates examples of the scheduler, task and machine stevedore, failures loader, and additional entities.

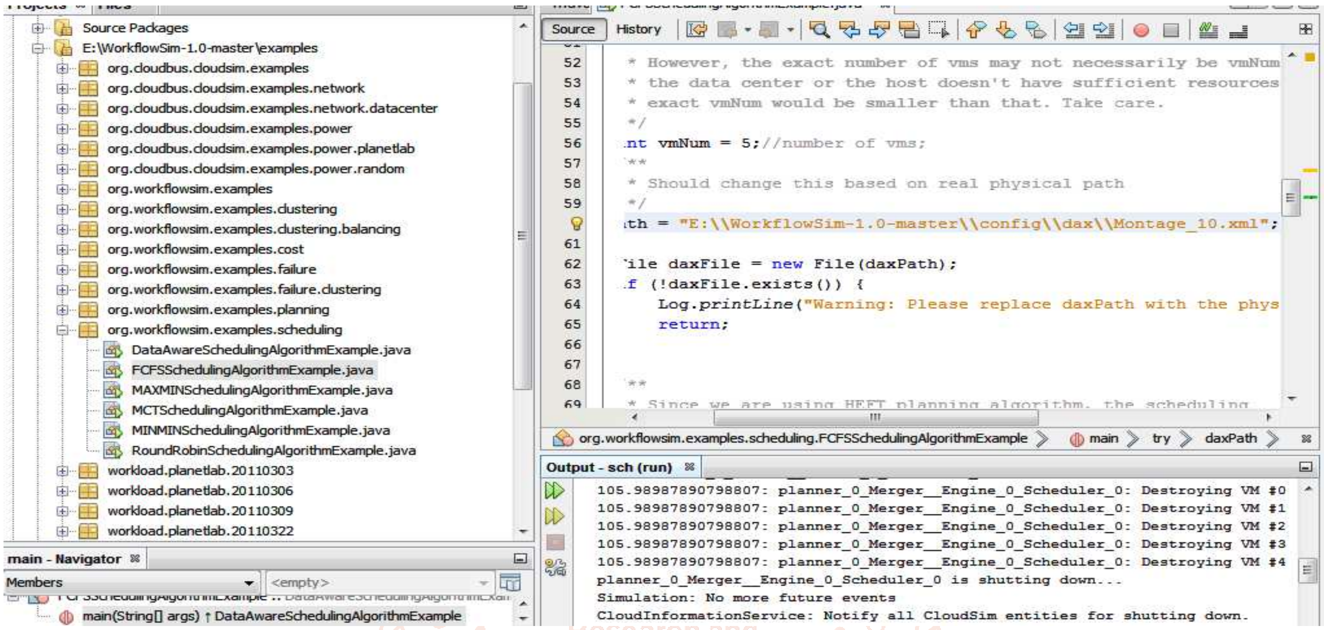

Figure 1: NetBeans IDE Environment with CloudSim 3.0.2 Environment

The following is the proposed method for evaluating the makespan (in ms) using FCFS, MCT, and ETRCS:

Table 1: FCFS, MCT, and ETRCS Makespan Comparison

Jobs	MAKESPAN		
	FCFS	MCT	ETRCS (Proposed)
2	119.46	132.53	72.1
4	267.52	210.33	151.68
6	472.36	470.45	379.04
8	527.32	541.68	465.9
10	928.61	925.08	710.65

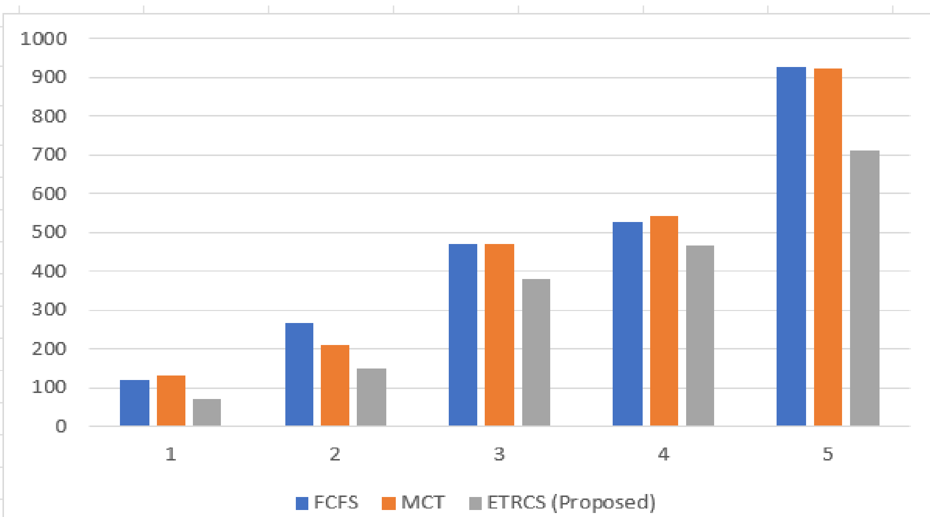

Figure 2: ETRCS, MCT, and FCFS Makespan

According to the graphical study above, ETRCS has a shorter makespan than FCFS and MCT. As a result, ETRCS is superior than MCT and FCFS. The following is the proposed method for evaluating the Minimum Scheduling Execution Time (in seconds) using FCFS, MCT, and ETRCS:

Table 2: FCFS, MCT, and ETRCS Minimum Scheduling Execution Time (sec) Comparison

Jobs	MSET		
	FCFS	MCT	ETRCS (Proposed)
2	0.15	0.12	0.11
4	0.17	0.19	0.12
6	0.67	0.8	0.2
8	0.73	0.87	0.23
10	0.88	1.01	0.54

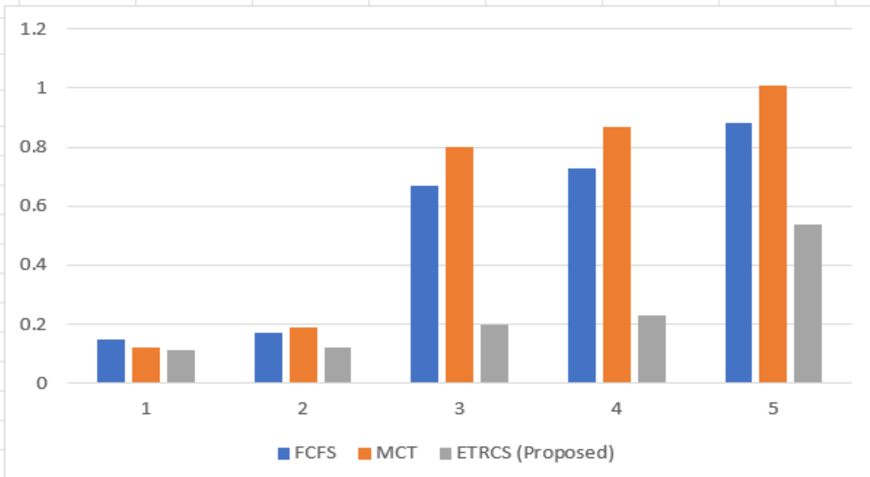

Figure 3: FCFS, MCT, and ETRCS MSET (sec)

According to the graphical study above, ITMCS has a lower Minimum Scheduling Execution Time than FCFS and MCT. As a result, ETRCS is superior than MCT and FCFS. The following is how the Supply Consumption Amount (in per) may be assessed using FCFS policy, MCT policy, and ETRCS (Projected):

Table 3: FCFS, MCT, and ETRCS Resource Utilization Rate Comparison (in per)

Jobs	RUR		
	FCFS	MCT	ETRCS
2	87.11	96.6	97.2
4	77.32	73.33	82.05
6	60.06	58.17	77.12
8	41.4	38.71	58.07
10	27.96	24.07	30.95

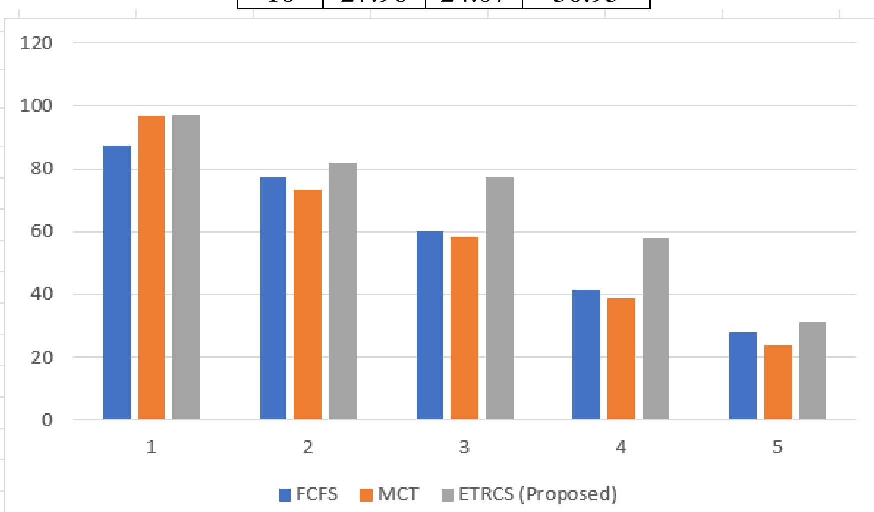

Figure 4: Resource Utilization Rate (sec) for FCFS, MCT, and ETRCS.

According to the graphical study above, ETRCS has a higher resource utilization rate (in per) than FCFS and MCT. As a result, ETRCS is superior than MCT and FCFS. The following is how FCFS, MCT, and ETRCS (Proposed Method) may be used to assess the skewness of Makespan (SM) and Makespan Standard Deviation (MSD) for the MT (Montage) and CS (Cybershake) dataset:

Table 4: Makespan and Makespan Standard Deviation Comparison for FCFS, MCT, and ETRCS

Scheduling Policy	MT (Montage)		CS (Cybershake)	
	Makespan	Makespan Standard Deviation	Makespan	Makespan Standard Deviation
First Come First Serve	2.3	29.11	17.35	149.81
Minimum Completion Time	3.62	42.21	14.7	121.33
Enhanced Task Relocation Consolidation Scheduling	2.05	28.72	12.31	120.16

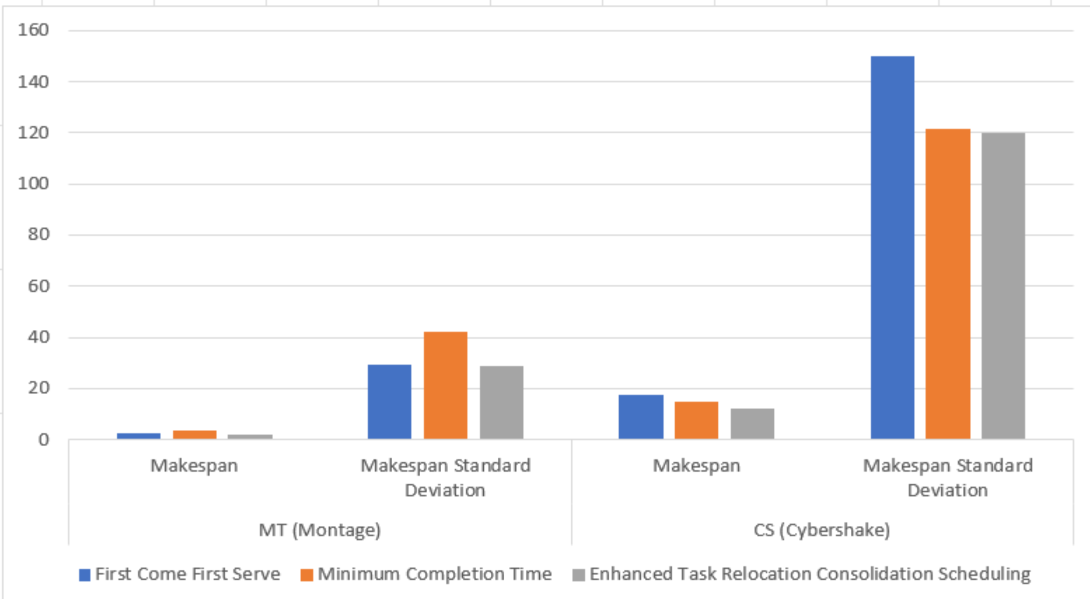

Figure 5: SM and MSD among FCFS, MCT, and ETRCS

According to the graphical analysis above, ETRCS has lower SM and MSD than FCFS and MCT. As a result, ETRCS is superior than MCT and FCFS.

References

[1] Babar, M. I., Khan, M. T., & Akbar, R. (2022). Energy-aware dynamic scheduling for cloud data centers. *Journal of Cloud Computing: Advances, Systems and Applications, 11*(1), 45-60.

[2] Bose, R., Dey, D., & Ghosh, S. (2021). Multi-objective optimization for job scheduling in cloud computing: A comprehensive survey. *Journal of Parallel and Distributed Computing, 153*, 1-15.

[3] Hussain, A., Malik, S. U., & Lee, H. (2021). Hybrid machine learning-based job scheduling in cloud environments. *Computing, 103*(9), 2247-2262.

[4] Kumar, S., & Soni, P. (2023). Green computing strategies for energy-efficient job scheduling in cloud environments. *Future Generation Computer Systems, 118*, 303-317.

[5] Liu, J., Zhang, Y., & Zhao, Z. (2021). A deep reinforcement learning approach for efficient resource allocation in cloud computing. *Future Internet, 13*(12), 332-345.

[6] Rao, R., Venkatesh, B., & Yegnanarayana, B. (2023). A dynamic load balancing algorithm for cloud computing systems. *Journal of Cloud Computing, 12*(2), 121-134.

[7] Singh, R., Gupta, N., & Sharma, M. (2022). Hybrid job scheduling algorithms for cloud and edge computing systems. *Cloud Computing and Big Data, 9*(3), 56-72.

[8] Zhang, L., Luo, Z., & Xu, W. (2023). A game theory-based load balancing approach for cloud computing systems. *IEEE Transactions on Cloud Computing, 11*(4), 1186-1197.

[9] Zhou, L., Wu, F., & Zhang, H. (2022). Metaheuristic-based job scheduling algorithms in cloud computing environments: A survey. *International Journal of Cloud Computing and Services Science, 11*(1), 45-61.